UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/567,822 | 02/08/2006 | Jinzhan Peng | 42P22630 | 4416 |

59796 7590 05/11/2009
INTEL CORPORATION
c/o CPA Global
P.O. BOX 52050
MINNEAPOLIS, MN 55402

| EXAMINER | |
|---|---|
| PETRANEK, JACOB ANDREW | |
| ART UNIT | PAPER NUMBER |
| 2183 | |
| MAIL DATE | DELIVERY MODE |
| 05/11/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| *Office Action Summary* | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/567,822 | PENG ET AL. | |
| | **Examiner** | **Art Unit** | |
| | Jacob Petranek | 2183 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on _17 February 2009_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) _1-20_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-20_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-20 are pending.

2. The office acknowledges the following papers:

Claims and arguments filed on 2/17/2009.

### *Withdrawn objections and rejections*

3. The specification objections have been withdrawn.

4. The 35 U.S.C. 101 rejections for claims 12-20 have been withdrawn due to amendment.

5. The 35 U.S.C. 112 second paragraph rejections for claims 7 and 9-10 are withdrawn due to amendment.

### *Maintained Claim Rejections - 35 USC § 102*

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 6-11 are rejected under 35 U.S.C. §102(b) as being anticipated by Adl-Tabatabai et al. (U.S. 6,317,869).

8. As per claim 6:

Adl-Tabatabai disclosed a method for managing type information for operands, the method comprising:

shifting a bit value of 1 into a register, in conjunction with creation of a reference operand (Adl-Tabatabai: Figures 4b and 5b element 555, column 6 lines 39-56)(Shifting can be defined as switching or exchange of. Thus, the replacement of data in the bit vector with a bit value of 1 is shifting.); and

shifting a bit value of 0 into the register, in conjunction with creation of a non-reference operand (Adl-Tabatabai: Figures 4b and 5b element 550, column 6 lines 39-56)(Shifting can be defined as switching or exchange of. Thus, the replacement of data in the bit vector with a bit value of 0 is shifting.).

9. As per claim 7:

Adl-Tabatabai disclosed a method according to claim 6, wherein:

the register serves as a tag stack register, the tag stack register to be used for storing a stack of operand tags (Adl-Tabatabai: Figure 4b, column 6 lines 8-16 and 25-28), each operand tag to indicate whether a corresponding operand on an operand stack is to be treated as a reference operand or a non-reference operand (Adl-Tabatabai: Figure 4b, column 6 lines 8-16)(The bits indicate if an operand makes a reference to an object or not.); and

the method further comprises initializing the tag stack register by:

assigning a low order bit of the tag stack register to a value of 0 (Adl-Tabatabai: Figure 5a element 515, column 6 lines 8-16); and

assigning other bits of the tag stack register to a value of 1 (Adl-Tabatabai: Figure 5a element 515, column 6 lines 8-16).

10. As per claim 8:

Adl-Tabatabai disclosed a method according to claim 6, further comprising:

using shift left operations to shift bit values into a low order bit of the register in response to operands being added to an operand stack (Adl-Tabatabai: Figures 4b and 5b element 555, column 6 lines 39-56)(Shifting can be defined as switching or exchange of. Thus, the replacement of data in the bit vector with a bit value of 1 is shifting. Adding a value to the low order bit in the bit vector reads upon the limitation.).

11. As per claim 9:

Adl-Tabatabai disclosed a method according to claim 6, further comprising:

right shifting bit values in the register in conjunction with removal of an operand (Adl-Tabatabai: Figures 4b and 5b element 550, column 6 lines 39-56)(Shifting can be defined as switching or exchange of. Thus, the replacement of data in the bit vector with a bit value of 0 is shifting.), the operand being one of the reference operand and the non-reference operand (Adl-Tabatabai: Column 6 lines 49-57)(The variable is inherently either referenced or non-referenced.).

12. As per claim 10:

Adl-Tabatabai disclosed a method according to claim 9, further comprising:

shifting the bit value of 1 into a high order bit of the register in conjunction with removal of the operand (Adl-Tabatabai: Figures 4b and 5b element 555, column 6 lines 39-56)(Shifting can be defined as switching or exchange of. Thus, the replacement of data in the bit vector with a bit value of 1 is shifting. Adding a value to the high order bit in the bit vector reads upon the limitation.).

13. As per claim 11:

Adl-Tabatabai disclosed a method according to claim 6, wherein:

the register serves as a tag stack register, the tag stack register to be used for storing a stack of operand tags (Adl-Tabatabai: Figure 4b, column 6 lines 8-16 and 25-28), each operand tag to indicate whether a corresponding operand on an operand stack is to be treated as a reference operand or a non-reference operand (Adl-Tabatabai: Figure 4b, column 6 lines 8-16)(The bits indicate if an operand makes a reference to an object or not.); and

the method further comprises:

treating a highest order bit with the value of 0 in the tag stack register as a stack pointer (Adl-Tabatabai: Figure 5a element 525, column 6 lines 28-38); and

determining a depth of the stack of operand tags, based at least in part on a location of the stack pointer (Adl-Tabatabai: Figure 5a element 525, column 6 lines 28-38).

## *New Claim Rejections - 35 USC § 103*

14. The following is a quotation of 35 U.S.C. §103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 1-5 and 12-20 are rejected under 35 U.S.C. §103(a) as being unpatentable over Ramesh et al. (U.S. 6,651,159).

16. As per claim 1:

Ramesh disclosed a method for managing type information for operands, the method comprising:

accomplishing the following results through execution of a single register instruction in a register of a processor (Ramesh: Figure 3 element 30, column 3 lines 25-49)(A single pop or push instruction uses a single register to add or delete data from the stack in the processor.):

adding an operand tag to a tag stack (Ramesh: Figure 4 element 33, column 3 lines 32-37 and column 4 lines 57-64)(When a push operation occurs, a tag for the operand is added to the pseudo-tag register.); and

updating a stack pointer for the tag stack to recognize the addition of the operand tag to the tag stack (Ramesh: Figure 5B, column 5 lines 55-63 continued to column 6 lines 1-6)(The top of stack pointer is updated when data is added or deleted to the stack. It's obvious to one of ordinary skill in the art that a single instruction performs both the updating of the stack pointers and adding/deleting data to/from the stack and pseudo-tag register.).

17. As per claim 2:

Ramesh disclosed a method according to claim 1, wherein the single register instruction comprises a shift instruction (Ramesh: Figure 5B, column 5 lines 55-63 continued to column 6 lines 1-6)(Data is shifted into the pseudo-tag register.).

18. As per claim 3:

Ramesh disclosed a method according to claim 2, wherein the shift instruction comprises a rotate instruction (Ramesh: Figure 5B, column 5 lines 55-63 continued to column 6 lines 1-6)(Data is shifted into the pseudo-tag register. A shift and rotate operation are one in the same when the rotate operation doesn't wrap around, which isn't claimed.).

19. As per claim 4:

Ramesh disclosed a method according to claim 1, further comprising:

accomplishing the following results through execution of one register instruction (Ramesh: Figure 3 element 30, column 3 lines 25-49)(A single pop or push instruction uses a single register to add or delete data from the stack in the processor.):

removing an operand tag from the tag stack (Ramesh: Figure 4 element 33, column 3 lines 38-49 and column 4 lines 57-64)(When a pop operation occurs, a tag for the operand is deleted from the pseudo-tag register.); and

updating the stack pointer for the tag stack to recognize the removal of the operand tag from the tag stack (Ramesh: Figure 5B, column 5 lines 55-63 continued to column 6 lines 1-6)(The top of stack pointer is updated when data is added or deleted to the stack. It's obvious to one of ordinary skill in the art that a single instruction performs both the updating of the stack pointers and adding/deleting data to/from the stack and pseudo-tag register.).

20. As per claim 5:

Ramesh disclosed a method according to claim 4, wherein the one register instruction comprises a shift right instruction (Ramesh: Figure 5B, column 5 lines 55-63 continued to column 6 lines 1-6)(Data is right-shifted into the pseudo-tag register when a pop instruction occurs.).

21. As per claim 12:

Ramesh disclosed a processing system with control logic for managing type information for operands, the processing system comprising:

a processor (Ramesh: Figure 1, column 2 lines 40-42);

a machine-accessible storage medium responsive to the processor (Ramesh: Figure 1)(The instruction executed by Ramesh are inherently stored within a memory within the processor of Ramesh.); and

instructions in the machine-accessible storage medium, the instructions to implement at least part of a virtual machine when executed by a processing system (Ramesh: Figure 1, column 2 lines 40-42)(The advantage of Java is that it's platform independent and is densely coded for compact programs to save memory space and power consumption. Official notice is given that processors can execute Java instructions. Thus, it's obvious to one of ordinary skill in the art to allow for the processor of Ramesh to execute Java instructions by implementing a virtual machine.), the virtual machine to accomplishing the following results through execution of a single register instruction (Ramesh: Figure 3 element 30, column 3 lines 25-49)(A single pop or push instruction uses a single register to add or delete data from the stack in the processor.):

adding an operand tag to a tag stack (Ramesh: Figure 4 element 33, column 3 lines 32-37 and column 4 lines 57-64)(When a push operation occurs, a tag for the operand is added to the pseudo-tag register.); and updating a stack pointer for the tag stack to recognize the addition of the operand tag to the tag stack (Ramesh: Figure 5B, column 5 lines 55-63 continued to column 6 lines 1-6)(The top of stack pointer is updated when data is added or deleted to the stack. It's obvious to one of ordinary skill in the art that a single instruction performs both the updating of the stack pointers and adding/deleting data to/from the stack and pseudo-tag register.).

22. As per claim 13:

The additional limitation(s) of claim 13 basically recite the additional limitation(s) of claim 2. Therefore, claim 13 is rejected for the same reason(s) as claim 2.

23. As per claim 14:

The additional limitation(s) of claim 14 basically recite the additional limitation(s) of claim 3. Therefore, claim 14 is rejected for the same reason(s) as claim 3.

24. As per claim 15:

The additional limitation(s) of claim 15 basically recite the additional limitation(s) of claim 4. Therefore, claim 15 is rejected for the same reason(s) as claim 4.

25. As per claim 16:

Ramesh disclosed a processing system according to claim 12 wherein the processor supports a little-endian byte order (Official notice is given that processors store data in either little-endian or big-endian byte order. Thus, it’s obvious to one of ordinary skill in the art that the system of Sokolov stores data in little-endian byte order.).

26. As per claim 17:

Claim 17 essentially recites the same limitations of claim 12. Therefore, claim 17 is rejected for the same reasons as claim 12.

27. As per claim 18:

The additional limitation(s) of claim 18 basically recite the additional limitation(s) of claim 2. Therefore, claim 18 is rejected for the same reason(s) as claim 2.

28. As per claim 19:

The additional limitation(s) of claim 19 basically recite the additional limitation(s) of claim 3. Therefore, claim 19 is rejected for the same reason(s) as claim 3.

29. As per claim 20:

The additional limitation(s) of claim 20 basically recite the additional limitation(s) of claim 4. Therefore, claim 20 is rejected for the same reason(s) as claim 4.

## *Response to Arguments*

30. The arguments presented by Applicant in the response, received on 2/17/2009 are partially considered persuasive.

31. Applicant argues "Claims 1-5 were rejected under 35 USC § 102(b) as being anticipated by Sokolov et al. (US Publication No. 2004/0015873). Applicant has amended claims 1, 2, 4, and 5 and respectfully submits that claims 1-5 are in condition for allowance. Applicant respectfully requests that claims 1-5 be allowed to pass to issuance."

This argument is found to be persuasive for the following reason. The examiner agrees that the amendment overcomes the cited prior art. However, a new ground of rejection has been given to the claims due to the amendment.

32. Applicant argues “The Office Action further explains that "shifting can be defined as switching or exchange of. Thus, the replacement of data in the bit vector with a bit value of 1 is shifting." Applicants respectfully disagree. One of skill in the art understands that shifting a bit value into a register performs an arithmetic operation on the contents of the register, thereby replacing the contents of the register with a different value. The values of all bits in the register may be affected by the shift operation. The setting of a single bit within a vector does not affect other bits within the vector, and thus Adl-Tabatabai does not teach "shifting a bit value [of 1 or 0] into a register."” for claim 6.

This argument is not found to be persuasive for the following reason. The applicant is correct in stating that a shifting operation performs an arithmetic operation on the contents of the register, and replacing the contents of the register with a different value. However, shifting that exchanges a bit within the vector also results in an arithmetic operation being performed with can result in a different overall value within a register.

In addition, placing a bit into the most and least significant bits of the bit vector can read upon shifting, since the data is shifted into that least or most significant bit position.

33. Applicant argues “Initialization of the bit vector is not discussed and would be irrelevant, since operations are performed only on single bits within the bit vector and

not on the bit vector as a whole. In contrast, initialization of the low order bit of the register to a value of 0 and other bits in the register to a value of 1 affects the contents of the register and enables the register to perform the arithmetic operation described above when a shift operation is performed" for claim 7.

This argument is not found to be persuasive for the following reason. A bit vector is established and initialized at compilation time and indicates the references to the variables. Inherently, if the all variables of the bit vector are referenced except for the lowest order bit variable, then the bit vector is initialized as having a zero in the low order bit position and a one in all other bit positions.

34. Applicant argues "Claims 12-20 were rejected under 35 USC § 103(a) as being unpatentable over Sokolov et al. (US Publication No. 2004/0015873). Applicant has amended claims 12, 13, 15, 17, 18, and 20, and respectfully submits that claims 12-20 are in condition for allowance. Applicant respectfully requests that claims 12-20 be allowed to pass to issuance."

This argument is found to be persuasive for the following reason. The examiner agrees that the amendment overcomes the cited prior art. However, a new ground of rejection has been given to the claims due to the amendment.

## *Conclusion*

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Banning et al. (U.S. 7,111,096), taught a tag holding register in a stack-based processor system.

Scheuneman (U.S. 4,757,440), taught utilizing stack register tag identifiers.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob Petranek whose telephone number is 571-272-5988. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

| | |
|---|---|
| /Eddie P Chan/<br>Supervisory Patent Examiner, Art Unit 2183 | Jacob Petranek<br>Examiner, Art Unit 2183 |